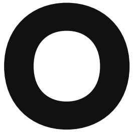




Programação do jogo



O papel do programador é desenvolver o software para criar o funcionamento dos jogos. Isso envolve a montagem das plataformas e mecanismos que alimentarão o projeto, bem como escrever o código fonte para suportar os requisitos exclusivos dos jogos.

Os programadores de jogos trabalham em estreita colaboração com designers e desenvolvedores durante o processo de pipeline do projeto, configurando o mecanismo e garantindo que a produção seja executada sem problemas. Uma grande parte de seu papel é a solução de problemas, portanto, um conhecimento sólido do software é necessário para ajudar a dar vida ao jogo.

A criação de protótipos que funcionarão como uma prova de conceito é um estágio inicial e crucial do pipeline de produção. Os programadores obterão uma melhor compreensão dos limites do jogo e como resolver os bugs ao longo do processo.

De acordo com ARRUDA (2014), os designers e desenvolvedores contam com os programadores para entregar a melhor versão do jogo possível para prosperar em mercados competitivos. Dentre os papéis e responsabilidades de um programador de jogos na indústria, podemos citar as seguintes tarefas:

- **Contato com designers e desenvolvedores de jogos na configuração de recursos técnicos.**

- Pesquisa e desenvolvimento do conjunto de software e plataformas que serão usados para dar suporte ao jogo.
- Garantir que o design do jogo seja totalmente realizado e executado na capacidade máxima.
- Criação de procedimentos e documentos de produção.
- Trabalhar com horários apertados e manter o orçamento.
- Produção de protótipos nos estágios iniciais de produção.
- Realização de testes de garantia de qualidade e resposta a feedback.
- Colaborar com todos os departamentos para resolver problemas técnicos, resolver problemas e criar soluções durante o processo de pipeline de produção.
- Responder às necessidades técnicas de todos os departamentos.
- Trabalhar em equipe em busca de objetivos comuns.
- Fornecer suporte técnico contínuo após o lançamento do jogo, além de trabalhar em atualizações para o jogo.

1.Programação em rede com sockets

Os chamados soquetes fornecem uma interface elementar para programar aplicativos de rede. Como uma API, os soquetes fornecem funções para estabelecer conexões e comunicação entre aplicativos. Eles encapsulam os detalhes das comunicações de rede. O programador

pode, assim, concentrar-se na realização da aplicação. No caso ideal, os aplicativos funcionam independentemente da linguagem de programação, sistema operacional e interface de rede (RABIN, 2012).

A API Socket é suportada por todos os sistemas operacionais comuns. Os soquetes são baseados nos protocolos de uma camada de transporte, como TCP ou UDP. Do ponto de vista do aplicativo, um soquete é um gateway para a rede.

A API Socket fornece dois métodos: comunicação orientada à conexão e sem conexão. No primeiro caso, os dois computadores envolvidos devem primeiro estabelecer uma conexão antes de poderem trocar dados. Em ambos os casos, no entanto, um dos dois computadores - o servidor - assume a tarefa de configurar inicialmente o recurso de rede. Em seguida, aguarda solicitações de conexão do outro computador - o cliente. Um servidor geralmente pode atender a vários clientes.

Os soquetes não são adequados apenas para comunicação entre computadores. Você também pode cuidar da comunicação entre vários aplicativos em um computador.

2. Programação Cliente Servidor

O modelo cliente-servidor ou arquitetura cliente-servidor é uma estrutura de aplicação distribuída que divide tarefas entre servidores e clientes, que residem no mesmo sistema ou se comunicam por meio de uma rede de computadores ou da Internet.

O cliente depende do envio de uma solicitação a outro programa para acessar um serviço disponibilizado por um servidor. Ele executa um ou mais programas que compartilham recursos e distribuem trabalho entre os clientes.

O relacionamento cliente-servidor se comunica em um padrão de mensagens de solicitação-resposta e deve aderir a um protocolo de comunicação comum, que define formalmente as regras, a linguagem e os padrões de diálogo a serem usados. A comunicação cliente-servidor normalmente segue o conjunto de protocolos TCP/IP (ARRUDA, 2014).

O protocolo TCP mantém uma conexão até que o cliente e o servidor concluam a troca de mensagens. O protocolo TCP determina a melhor maneira de distribuir dados de aplicativos em pacotes que as redes podem entregar, transfere e recebe pacotes da rede e gerencia o controle de fluxo e a retransmissão de pacotes perdidos ou distorcidos.

O IP é um protocolo sem conexão no qual cada pacote que viaja pela Internet é uma unidade independente de dados não relacionada a nenhuma outra unidade de dados.

As solicitações dos clientes são organizadas e priorizadas em um sistema de agendamento, o que ajuda os servidores a lidar com o recebimento de solicitações de muitos clientes distintos em um curto espaço de tempo. A abordagem cliente-servidor permite que qualquer computador de uso geral expanda seus recursos utilizando os recursos compartilhados de outros hosts. Os aplicativos cliente-servidor populares incluem e-mail, World Wide Web e impressão em rede.

A partir desse contexto, de acordo com RABIN (2012), podemos apontar quatro categorias principais de computação cliente-servidor:

- **Arquitetura One-Tier: consiste em um programa simples executado em um único computador sem a necessidade de acesso** à rede. **As solicitações do usuário não gerenciam nenhum protocolo de rede**. P**ortanto, o código é simples e a rede é liberada do tráfego extra.**

- **Arquitetura de duas camadas: consiste no cliente, no servidor e no protocolo que liga as duas camadas. O código da interface gráfica do usuário reside no**

host do cliente e a lógica do domínio reside no host do servidor. A GUI cliente-servidor é escrita em linguagens de alto nível, como C++ e Java.



- **Arquitetura de três camadas:** consiste em uma camada de apresentação, que é a camada de interface do usuário. A camada de aplicativo é a camada de serviço que realiza o processamento detalhado. Além disso, temos a camada de dados, que consiste em um servidor de banco de dados que armazena informações.

- **Arquitetura N-Tier:** divide um aplicativo em camadas lógicas, separando responsabilidades e gerenciando dependências. Além de camadas físicas que são executadas em máquinas separadas, melhorando a escalabilidade e adicionando latência da comunicação de rede adicional. A arquitetura N-Tier pode ser de camada fechada, em que uma camada só pode se comunicar com a próxima camada abaixo ou camada aberta, quando uma camada pode se comunicar com quaisquer camadas abaixo dela.

3. Programação de Threads (controle de rede)

A programação de Threads é um processo leve que executa a sequência de código e todas as estruturas de suporte de dados, como recursos abertos, mapa de memória, pilha, etc.

Para um programa, um thread pode se dividir em mais de duas partes simultaneamente enquanto executa as tarefas. Existe uma diferença entre os processos de cada sistema operacional e outro, embora, geralmente, uma thread seja constituída dentro de um processo, sendo distintas no próprio processo e compartilhando recursos semelhantes.

Segundo RABIN (2012), convencionalmente, os atributos de processo e encadeamento são organizados em uma entidade individual conhecida como processos. Em sistemas convencionais, o processo de uma única

thread consiste em um conjunto de características. Já em sistemas de multithreaded suas características são divididas entre linhas e processos.

4. Mecânica para Jogos Multiplayer

Os aplicativos multijogador estão em toda parte, já que esse modo é um dos pilares dos jogos AAA e também de pequenas ofertas independentes. Além disso, atendem ao nosso desejo humano mais fundamental.

Com a adoção generalizada da Internet de alta velocidade, a indústria do entretenimento também mudou. Os videogames estão se tornando cada vez mais jogos de serviço e geralmente consistem apenas em um modo multiplayer.

4.1 Modo multijogador

Assim que houver mais de um jogador em uma cena, podemos falar em multiplayer. No entanto, a geração mais jovem também associa automaticamente a palavra multiplayer com a Internet (RABIN, 2012). No entanto, existe outra forma de experimentar o conteúdo virtual com várias pessoas:

- **Multijogador Online:** requer uma conexão com a Internet. Os jogadores se conectam a um servidor ou diretamente aos outros jogadores.
- **Multijogador offline:** sem conexão com a internet, ocorre localmente no dispositivo. Dependendo do jogo, é possível jogar um contra o outro.
- **Mesma tela:** um desses aplicativos era o Pong, por exemplo, você não só podia competir com um oponente controlado por computador, mas também com um

segundo jogador sentado ao seu lado.



- **Split-Screen:** cada jogador tem uma seção de imagem com o jogo acontecendo na tela. Alguns aplicativos possibilitam essa experiência com até quatro jogadores. Exemplos bem conhecidos são Need For Speed ou Halo.

- **Hotseat:** alguns jogos oferecem a opção dos jogadores se revezarem. Isso também é possível com apenas um controlador.

A estrutura de um aplicativo multijogador online é essencial para a experiência de jogo. A arquitetura é responsável pelo controle dos dados e os diferentes modelos possuem suas vantagens e desvantagens. São elas:

- **Cliente-Servidor:** centralizado, pois o servidor está no controle e os clientes individuais (jogadores) são os principais responsáveispor inserir comandos.

- **Peer to peer (P2P):** descentralizado, pois cada cliente individual controla suas ações individuais e as encaminha para os outros clientes. Um servidor não é necessário para isso.

- **Híbrido:** combinação, usando um cliente master e um servidor. O cliente mestre assume o papel do servidor e cuida da sincronização entre os clientes individuais conectados ao servidor. O servidor serve como intermediário para os pacotes de dados e também pode ser usado como órgão de controle.

Atividade Extra

A partir do que foi apresentado neste módulo, o trabalho acadêmico intitulado de "**A Aplicação do Framework SCRUM no Desenvolvimento Jogos Multiplayer Online na Academia: A Experiência com o Jogo Dyspair**" (PEREIRA et. al., 2019), disponível no Google Acadêmico, aborda sobre o processo de desenvolvimento de um jogo digital, a partir da utilização do Framework SCRUM como estratégia-base de gestão do projeto.

O projeto identificou os requisitos necessários para se desenvolver um jogo da categoria MOBA (multiplayer online battle arena) e traçou uma estratégia de gestão baseada nestas features, utilizando-se de equipes multidisciplinares, onde os resultados alcançados foram relatados e avaliados em relação a outros projetos.

Referência Bibliográfica

ARRUDA, E. P. **Fundamentos para o Desenvolvimento de Jogos Digitais**. Porto Alegre: Grupo A, 2014.

RABIN, S. **Introdução ao Desenvolvimento de Games**. São Paulo: Cengage Learning Brasil, 2012.

Atividade Prática

Programação do jogo

Título da Prática: Programação do jogo

Objetivos: Diferenciar sobre os tipos de estruturas de programação.

Materiais, Métodos e Ferramentas: Browser (navegador) e ferramenta de edição de textos.

Atividade Prática



Roteiro

1º. Passo) Leia atentamente o texto abaixo:

Na comparação entre designers versus programadores de jogos digitais, vemos que o trabalho dos programadores é muito mais técnico, pois eles lidam com códigos e outras linguagens para criar jogos divertidos para as pessoas. Os programadores devem ter conhecimento de matemática, lógica, programação de computadores, além da compreensão de várias linguagens de programação para criar jogos sofisticados e divertidos.

Os programadores de jogos são membros integrantes de uma equipe de desenvolvimento, pois criam os blocos que realizam a visão dos designers de jogos. De certa forma, eles são os construtores da indústria de jogos. Cada programador é responsável por garantir que os jogos sejam executados conforme projetados e previstos com o mínimo de problemas.

Devido ao alto nível de tecnicidade e dificuldade que os programadores enfrentam, eles têm um vasto conhecimento no aspecto técnico do desenvolvimento, particularmente em codificação e ciência da computação. Geralmente, eles têm uma especialização em termos de desenvolvimento de jogos, pois a programação pode ser bastante ampla e abrangente.

Os programadores podem ter algumas das seguintes subespecialidades

- programação gráfica;
- programação de rede;
- programação de IA.

A programação de videogames não é fácil. É por isso que, na maioria das empresas de jogos, os programadores são muito procurados. Ser capaz transformar o design e as ideias em jogos reais exige habilidade e esforço, tornando esses programadores uma parte valiosa e insubstituível da indústria de jogos.

2º. Passo) Leia o trabalho acadêmico intitulado de "WebSockets e a Sua Aplicação no Mundo Web" (Almeida, 2019), disponível no Google Acadêmico.

A linguagem de programação Java em seu estado atual ainda é um pouco lenta, o que o descarta a sua utilização como sendo uma possibilidade para o desenvolvimento de jogos de alto desempenho e graficamente intensos. No entanto, o Java certamente pode ser usado para outros tipos de jogos, como jogos de estratégia multiplayer.

Um dos principais pontos fortes do Java como linguagem de programação é sua ampla gama de suporte de rede. A linguagem Java tem essa vantagem, porque foi desenvolvida pensando na Internet. O resultado é que você tem muitas opções em relação à programação de rede em Java. Embora existam muitas opções de rede, a programação de jogos em rede utilizando a linguagem Java usa um tipo específico de comunicação de rede conhecido como soquetes. Um soquete é uma abstração de software para um meio de comunicação de entrada ou saída.

O Java executa toda a sua comunicação de rede de baixo nível por meio de soquetes. Logicamente, os soquetes estão um passo abaixo das portas; você usa eles para se comunicar por meio de uma porta específica. Portanto, um soquete é um canal de comunicação que permite transferir dados por meio de uma determinada porta.

3º. Passo) A partir desse contexto, pesquise sobre a Programação Socket em

Redes utilizando a linguagem de programação Java no processo de desenvolvimento de jogos digitais.



Na atividade você deverá construir um relatório apontando algumas características e as vantagens da Programação Socket em Redes, utilizando a linguagem de programação Java no processo de desenvolvimento de jogos digitais.

Você deverá enviar o seu relatório técnico seguindo o formato ABNT, contendo entre 3000 e 3500 caracteres (com espaços) e desconsiderando as referências bibliográficas. Na montagem desse documento, será permitido a utilização de imagens, gráficos e demais elementos para facilitar a compreensão da sua ideia.

Gabarito Atividade Prática

A linguagem de programação Java fornece classes de soquete para tornar a programação muito mais fácil. Os soquetes Java são divididos em dois tipos: soquetes de fluxo e soquetes de datagrama.

Um soquete de fluxo ou conectado é um soquete pelo qual os dados podem ser transmitidos continuamente. O benefício de usar um soquete de fluxo é que as informações podem ser enviadas com menos preocupação sobre quando chegarão ao seu destino. Como o link de comunicação está sempre “ativo”, os dados geralmente são transmitidos imediatamente após o envio.

A linguagem Java oferece suporte para a programação de soquete de fluxo principalmente por meio de duas classes: Socket e ServerSocket. O Socket fornece a sobrecarga necessária para facilitar um cliente de soquete de fluxo. Já o ServerSocket fornece a funcionalidade principal para um servidor. A maior parte do código real que facilita a comunicação por meio

de soquetes é manipulada por meio de fluxos de entrada e saída conectados a um soquete.



Dessa forma, a própria comunicação é tratada independentemente da conexão do soquete de rede. Isso pode não parecer grande coisa a princípio, mas é crucial no projeto das classes de soquete; depois de criar um soquete, podemos conectar um fluxo de entrada ou saída a ele. Outro tipo de soquete suportado pelo Java é o soquete de datagrama. Ao contrário dos soquetes de fluxo, em que a comunicação é semelhante a uma rede ao vivo, um soquete de datagrama é mais semelhante a uma rede dial-up, na qual o link de comunicação não está continuamente ativo. É um soquete pelo qual os dados são agrupados em pacotes e enviados sem a necessidade de uma conexão ao vivo com o computador de destino.

Tratando-se da natureza do meio de comunicação envolvido, não é garantido que os soquetes de datagrama transmitam informações em um determinado momento ou mesmo em qualquer ordem específica. A razão pela qual os soquetes de datagrama funcionam dessa maneira é que eles não requerem uma conexão real com outro computador; o endereço do computador de destino é apenas empacotado com as informações que estão sendo enviadas. Esse pacote é então enviado pela Internet, deixando o remetente esperando pelo melhor.

No lado receptor, os pacotes de informações podem ser recebidos em qualquer ordem e a qualquer momento. Por esse motivo, os datagramas também incluem um número de sequência que especifica qual peça do quebra-cabeça cada feixe corresponde. O receptor espera para receber toda a sequência e, em seguida, os reúne novamente para formar uma transferência completa de informações. Como você pode pensar, soquetes de datagrama não são ideais para a programação de jogos em rede, devido aos atrasos de tempo implícitos, problemas de confiabilidade elevados e complexidades de sequenciamento.



Ir para exercício